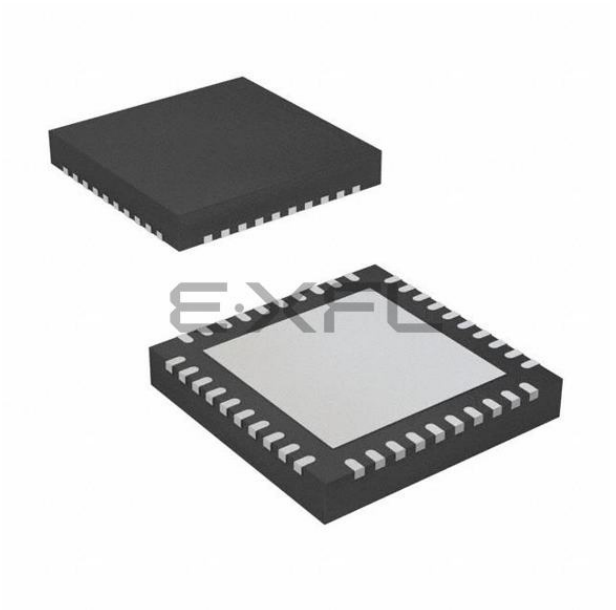

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	8051
Core Size	8-Bit
Speed	50MHz
Connectivity	EBI/EMI, SMBus (2-Wire/I²C), LINbus, SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	33
Program Memory Size	16KB (16K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.25V
Data Converters	A/D 32x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	40-VFQFN Exposed Pad
Supplier Device Package	40-QFN (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/silicon-labs/c8051f574-imr

2. Ordering Information

The following features are common to all devices in this family:

- 50 MHz system clock and 50 MIPS throughput (peak)
- 2304 bytes of RAM (256 internal bytes and 2048 XRAM bytes)
- SMBus/I^{2}C, Enhanced SPI, Enhanced UART
- Four Timers
- Six Programmable Counter Array channels
- Internal 24 MHz oscillator
- Internal Voltage Regulator
- 12-bit, 200 ksps ADC
- Internal Voltage Reference and Temperature Sensor
- Two Analog Comparators

Table 2.1 shows the feature that differentiate the devices in this family.

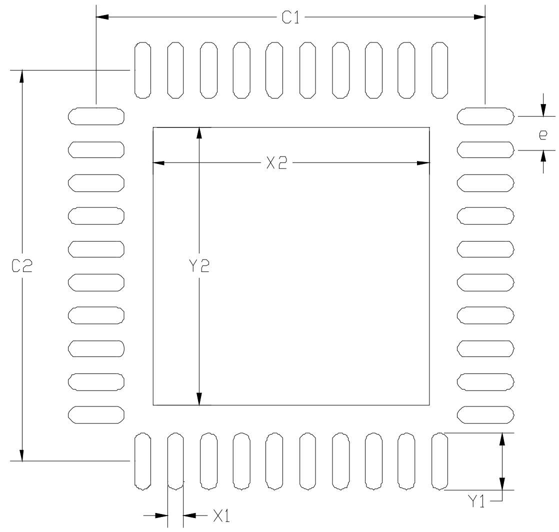

Figure 4.2. QFN-40 Landing Diagram

Table 4.2. QFN-40 Landing Diagram Dimensions

Dimension	Min	Max	Dimension	Min	Max
C1	5.80	5.90	X2	4.10	4.20
C2	5.80	5.90	Y1	0.75	0.85
e	0.50 BSC		Y2	4.10	4.20
X1	0.15	0.25			

Notes:

General

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. Dimension and Tolerancing is per the ANSI Y14.5M-1994 specification.
3. This Land Pattern Design is based on the IPC-SM-7351 guidelines.
4. All dimensions shown are at Maximum Material Condition (MMC). Least Material Condition (LMC) is calculated based on a Fabrication Allowance of 0.05 mm.

Solder Mask Design

5. All metal pads are to be non-solder mask defined (NSMD). Clearance between the solder mask and the metal pad is to be 60 µm minimum, all the way around the pad.

Stencil Design

6. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
7. The stencil thickness should be 0.125 mm (5 mils).
8. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pads.
9. A 4x4 array of 0.80 mm square openings on a 1.05 mm pitch should be used for the center ground pad.

Card Assembly

10. A No-Clean, Type-3 solder paste is recommended.
11. Recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

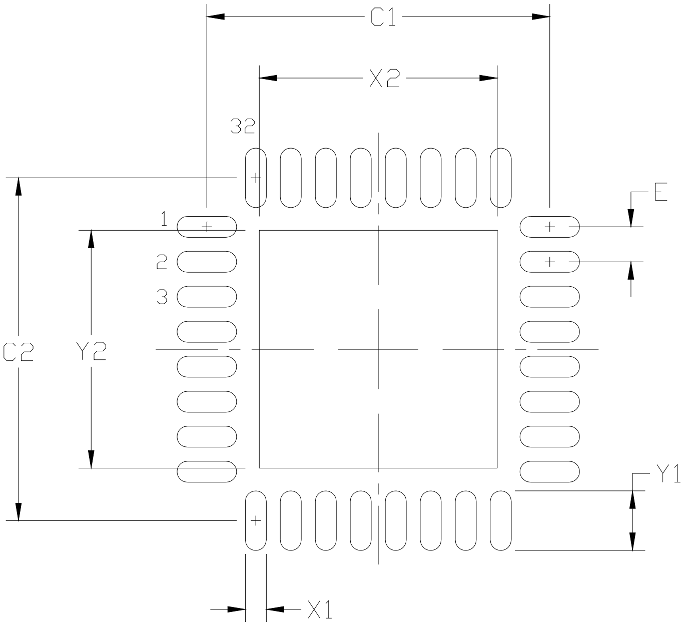

Figure 4.6. QFN-32 Landing Diagram

Table 4.6. QFN-32 Landing Diagram Dimensions

Dimension	Min	Max	Dimension	Min	Max
C1	4.80	4.90	X2	3.20	3.40
C2	4.80	4.90	Y1	0.75	0.85
e	0.50 BSC		Y2	3.20	3.40
X1	0.20	0.30			

Notes:

General

1. All dimensions shown are in millimeters (mm) unless otherwise noted.
2. This Land Pattern Design is based on the IPC-7351 guidelines.

Solder Mask Design

3. All metal pads are to be non-solder mask defined (NSMD). Clearance between the solder mask and the metal pad is to be 60 µm minimum, all the way around the pad.

Stencil Design

4. A stainless steel, laser-cut and electro-polished stencil with trapezoidal walls should be used to assure good solder paste release.
5. The stencil thickness should be 0.125 mm (5 mils).
6. The ratio of stencil aperture to land pad size should be 1:1 for all perimeter pads.
7. A 3x3 array of 1.0 mm openings on a 1.20 mm pitch should be used for the center ground pad.

Card Assembly

8. A No-Clean, Type-3 solder paste is recommended.
9. The recommended card reflow profile is per the JEDEC/IPC J-STD-020 specification for Small Body Components.

8. Comparators

The C8051F55x/56x/57x devices include two on-chip programmable voltage Comparators. A block diagram of the comparators is shown in Figure 8.1, where “n” is the comparator number (0 or 1). The two Comparators operate identically except that Comparator0 can also be used a reset source. For input selection details, refer to SFR Definition 8.5 and SFR Definition 8.6.

Each Comparator offers programmable response time and hysteresis, an analog input multiplexer, and two outputs that are optionally available at the Port pins: a synchronous “latched” output (CP0, CP1), or an asynchronous “raw” output (CP0A, CP1A). The asynchronous signal is available even when the system clock is not active. This allows the Comparators to operate and generate an output with the device in STOP mode. When assigned to a Port pin, the Comparator outputs may be configured as open drain or push-pull (see Section “19.4. Port I/O Initialization” on page 174). Comparator0 may also be used as a reset source (see Section “16.5. Comparator0 Reset” on page 142).

The Comparator0 inputs are selected in the CPT0MX register (SFR Definition 8.5). The CMX0P1-CMX0P0 bits select the Comparator0 positive input; the CMX0N1-CMX0N0 bits select the Comparator0 negative input. The Comparator1 inputs are selected in the CPT1MX register (SFR Definition 8.6). The CMX1P1-CMX1P0 bits select the Comparator1 positive input; the CMX1N1-CMX1N0 bits select the Comparator1 negative input.

Important Note About Comparator Inputs: The Port pins selected as Comparator inputs should be configured as analog inputs in their associated Port configuration register, and configured to be skipped by the Crossbar (for details on Port configuration, see Section “19.1. Port I/O Modes of Operation” on page 170).

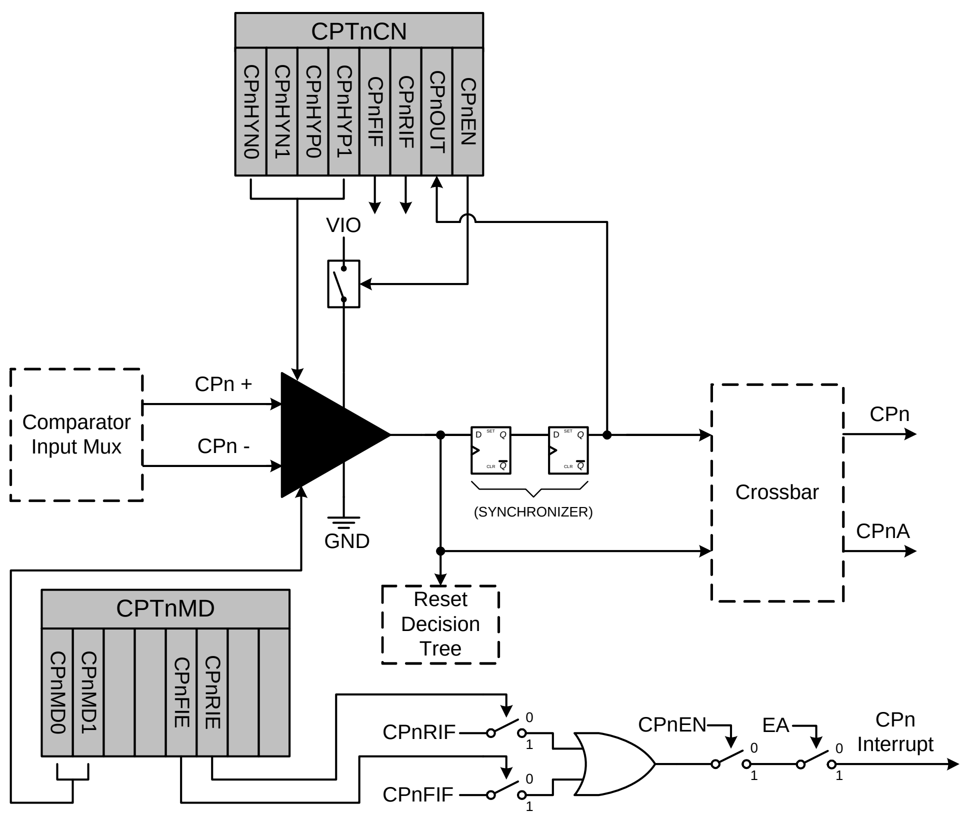

Figure 8.1. Comparator Functional Block Diagram

10. CIP-51 Microcontroller

The MCU system controller core is the CIP-51 microcontroller. The CIP-51 is fully compatible with the MCS-51™ instruction set; standard 803x/805x assemblers and compilers can be used to develop software. The MCU family has a superset of all the peripherals included with a standard 8051. The CIP-51 also includes on-chip debug hardware (see description in Section 27), and interfaces directly with the analog and digital subsystems providing a complete data acquisition or control-system solution in a single integrated circuit.

The CIP-51 Microcontroller core implements the standard 8051 organization and peripherals as well as additional custom peripherals and functions to extend its capability (see Figure 10.1 for a block diagram). The CIP-51 includes the following features:

- Fully Compatible with MCS-51 Instruction Set
- 50 MIPS Peak Throughput with 50 MHz Clock
- 0 to 50 MHz Clock Frequency
- Extended Interrupt Handler
- Reset Input
- Power Management Modes
- On-chip Debug Logic
- Program and Data Memory Security

10.1. Performance

The CIP-51 employs a pipelined architecture that greatly increases its instruction throughput over the standard 8051 architecture. In a standard 8051, all instructions except for MUL and DIV take 12 or 24 system clock cycles to execute, and usually have a maximum system clock of 12 MHz. By contrast, the CIP-51 core executes 70% of its instructions in one or two system clock cycles, with no instructions taking more than eight system clock cycles.

Table 12.2. Special Function Register (SFR) Memory Map for Page 0x0C

	0(8)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)
F8			CAN0IF2DA2L	CAN0IF2DA2H	CAN0IF2DB1L	CAN0IF2DB1H	CAN0IF2DB2L	CAN0IF2DB2H
F0	B (All Pages)		CAN0IF2A2L	CAN0IF2A2H			CAN0IF2DA1L	CAN0IF2DA1H
E8			CAN0IF2M1L	CAN0IF2M1H	CAN0IF2M2L	CAN0IF2M2H	CAN0IF2A1L	CAN0IF2A1H
E0	ACC (All Pages)		CAN0IF2CML	CAN0IF2CMH			EIE1 (All Pages)	EIE2 (All Pages)
D8			CAN0IF1DB1L	CAN0IF1DB1H	CAN0IF1DB2L	CAN0IF1DB2H	CAN0IF2CRL	CAN0IF2CRH
D0	PSW (All Pages)		CAN0IF1MCL	CAN0IF1MCH	CAN0IF1DA1L	CAN0IF1DA1H	CAN0IF1DA2L	CAN0IF1DA2H
C8			CAN0IF1A1L	CAN0IF1A1H	CAN0IF1A2L	CAN0IF1A2H	CAN0IF2MCL	CAN0IF2MCH
C0	CAN0CN		CAN0IF1CML	CAN0IF1CMH	CAN0IF1M1L	CAN0IF1M1H	CAN0IF1M2L	CAN0IF1M2H
B8	IP (All Pages)		CAN0MV1L	CAN0MV1H	CAN0MV2L	CAN0MV2H	CAN0IF1CRL	CAN0IF1CRH
B0	P3 (All Pages)		CAN0IP2L	CAN0IP2H		P4 (All Pages)	FLSCL (All Pages)	FLKEY (All Pages)
A8	IE (All Pages)		CAN0ND1L	CAN0ND1H	CAN0ND2L	CAN0ND2H	CAN0IP1L	CAN0IP1H
A0	P2 (All Pages)	CAN0BRPE	CAN0TR1L	CAN0TR1H	CAN0TR2L	CAN0TR2H		SFRPAGE (All Pages)
98	SCON0 (All Pages)		CAN0BTL	CAN0BTH	CAN0IIDL	CAN0IIDH	CAN0TST	
90	P1 (All Pages)		CAN0CFG		CAN0STAT		CAN0ERRL	CAN0ERRH
88	TCON (All Pages)	TMOD (All Pages)	TL0 (All Pages)	TL1 (All Pages)	TH0 (All Pages)	TH1 (All Pages)	CKCON (All Pages)	
80	P0 (All Pages)	SP (All Pages)	DPL (All Pages)	DPH (All Pages)		SFRNEXT (All Pages)	SFRLAST (All Pages)	PCON (All Pages)
	0(8) (bit addressable)	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

14. Flash Memory

On-chip, re-programmable Flash memory is included for program code and non-volatile data storage. The Flash memory can be programmed in-system, a single byte at a time, through the C2 interface or by software using the MOVX instruction. Once cleared to logic 0, a Flash bit must be erased to set it back to logic 1. Flash bytes would typically be erased (set to 0xFF) before being reprogrammed. The write and erase operations are automatically timed by hardware for proper execution; data polling to determine the end of the write/erase operation is not required. Code execution is stalled during a Flash write/erase operation. Refer to Table 5.5 for complete Flash memory electrical characteristics.

14.1. Programming The Flash Memory

The simplest means of programming the Flash memory is through the C2 interface using programming tools provided by Silicon Labs or a third party vendor. This is the only means for programming a non-initialized device. For details on the C2 commands to program Flash memory, see Section “27. C2 Interface” on page 300.

To ensure the integrity of Flash contents, it is strongly recommended that the on-chip V_{DD} Monitor be enabled in any system that includes code that writes and/or erases Flash memory from software. See Section 14.4 for more details. Before performing any Flash write or erase procedure, set the FLEWT bit in Flash Scale register (FLSCL) to 1. Also, note that 8-bit MOVX instructions cannot be used to erase or write to Flash memory at addresses higher than 0x00FF.

For –I (Industrial Grade) parts, parts programmed at a cold temperature below 0 °C may exhibit weakly programmed flash memory bits. If programmed at 0 °C or higher, there is no problem reading Flash across the entire temperature range of -40 °C to 125 °C. This temperature restriction does not apply to –A (Automotive Grade) devices.

14.1.1. Flash Lock and Key Functions

Flash writes and erases by user software are protected with a lock and key function. The Flash Lock and Key Register (FLKEY) must be written with the correct key codes, in sequence, before Flash operations may be performed. The key codes are: 0xA5, 0xF1. The timing does not matter, but the codes must be written in order. If the key codes are written out of order, or the wrong codes are written, Flash writes and erases will be disabled until the next system reset. Flash writes and erases will also be disabled if a Flash write or erase is attempted before the key codes have been written properly. The Flash lock resets after each write or erase; the key codes must be written again before a following Flash operation can be performed. The FLKEY register is detailed in SFR Definition 14.2.

SFR Definition 14.4. CCH0CN: Cache Control

Bit	7	6	5	4	3	2	1	0
Name	Reserved	Reserved	CHPFEN	Reserved	Reserved	Reserved	Reserved	CHBLKW
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	1	0	0	0	0	0

SFR Address = 0xE3; SFR Page = 0x0F

Bit	Name	Function
7:6	Reserved	Must Write 00b
5	CHPFEN	**Cache Prefect Enable Bit.** 0: Prefetch engine is disabled. 1: Prefetch engine is enabled.
4:1	Reserved	Must Write 0000b.
0	CHBLKW	**Block Write Enable Bit.** This bit allows block writes to Flash memory from firmware. 0: Each byte of a software Flash write is written individually. 1: Flash bytes are written in groups of two.

SFR Definition 14.5. ONESHOT: Flash Oneshot Period

Bit	7	6	5	4	3	2	1	0
Name					PERIOD[3:0]			
Type	R	R	R	R	R/W	R/W	R/W	R/W
Reset	0	0	0	0	1	1	1	1

SFR Address = 0xBE; SFR Page = 0x0F

Bit	Name	Function
7:4	Unused	Read = 0000b. Write = don't care.
3:0	PERIOD[3:0]	**Oneshot Period Control Bits.** These bits limit the internal Flash read strobe width as follows. When the Flash read strobe is de-asserted, the Flash memory enters a low-power state for the remainder of the system clock cycle. $FLASH_{RDMAX} = 5ns + (PERIOD \times 5ns)$

SFR Definition 18.3. OSCICRS: Internal Oscillator Coarse Calibration

Bit	7	6	5	4	3	2	1	0
Name		OSCICRS[6:0]						
Type	R	R/W						
Reset	0	Varies	Varies	Varies	Varies	Varies	Varies	Varies

SFR Address = 0xA2; SFR Page = 0x0F

Bit	Name	Function
7	Unused	Read = 0; Write = Don't Care
6:0	OSCICRS[6:0]	**Internal Oscillator Coarse Calibration Bits.** These bits determine the internal oscillator period. When set to 0000000b, the internal oscillator operates at its slowest setting. When set to 1111111b, the internal oscillator operates at its fastest setting. The reset value is factory calibrated to generate an internal oscillator frequency of 24 MHz.

SFR Definition 18.4. OSCIFIN: Internal Oscillator Fine Calibration

Bit	7	6	5	4	3	2	1	0
			OSCIFIN[5:0]					
Type	R	R	R/W					
Reset	0	0	Varies	Varies	Varies	Varies	Varies	Varies

SFR Address = 0x9E; SFR Page = 0x0F

Bit	Name	Function
7:6	Unused	Read = 00b; Write = Don't Care
5:0	OSCIFIN[5:0]	**Internal Oscillator Fine Calibration Bits.** These bits are fine adjustment for the internal oscillator period. The reset value is factory calibrated to generate an internal oscillator frequency of 24 MHz.

$$f = (KF) / (R \times V_{DD})$$

Equation 18.2. C Mode Oscillator Frequency

For example: Assume V_{DD} = 2.1 V and f = 75 kHz:

f = KF / (C x VDD)

0.075 MHz = KF / (C x 2.1)

Since the frequency of roughly 75 kHz is desired, select the K Factor from the table in SFR Definition 18.6 (OSCXCN) as KF = 7.7:

0.075 MHz = 7.7 / (C x 2.1)

C x 2.1 = 7.7 / 0.075 MHz

C = 102.6 / 2.0 pF = 51.3 pF

Therefore, the XFCN value to use in this example is 010b.

SFR Definition 19.1. XBR0: Port I/O Crossbar Register 0

Bit	7	6	5	4	3	2	1	0
Name	CP1AE	CP1E	CP0AE	CP0E	SMB0E	SPI0E	CAN0E	URT0E
Type	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xE1; SFR Page = 0x0F

Bit	Name	Function
7	CP1AE	**Comparator1 Asynchronous Output Enable.** 0: Asynchronous CP1 unavailable at Port pin. 1: Asynchronous CP1 routed to Port pin.
6	CP1E	**Comparator1 Output Enable.** 0: CP1 unavailable at Port pin. 1: CP1 routed to Port pin.
5	CP0AE	**Comparator0 Asynchronous Output Enable.** 0: Asynchronous CP0 unavailable at Port pin. 1: Asynchronous CP0 routed to Port pin.
4	CP0E	**Comparator0 Output Enable.** 0: CP0 unavailable at Port pin. 1: CP0 routed to Port pin.
3	SMB0E	**SMBus I/O Enable.** 0: SMBus I/O unavailable at Port pins. 1: SMBus I/O routed to Port pins.
2	SPI0E	**SPI I/O Enable.** 0: SPI I/O unavailable at Port pins. 1: SPI I/O routed to Port pins. Note that the SPI can be assigned either 3 or 4 GPIO pins.
1	CAN0E	**CAN I/O Output Enable.** 0: CAN I/O unavailable at Port pins. 1: CAN_TX, CAN_RX routed to Port pins P0.6 and P0.7.
0	URT0E	**UART I/O Output Enable.** 0: UART I/O unavailable at Port pin. 1: UART TX0, RX0 routed to Port pins P0.4 and P0.5.

SFR Definition 19.21. P2MDIN: Port 2 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	P2MDIN[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xF3; SFR Page = 0x0F

Bit	Name	Function
7:0	P2MDIN[7:0]	**Analog Configuration Bits for P2.7–P2.0 (respectively).** Port pins configured for analog mode have their weak pull-up and digital receiver disabled. For analog mode, the pin also needs to be configured for open-drain mode in the P2MDOUT register. 0: Corresponding P2.n pin is configured for analog mode. 1: Corresponding P2.n pin is not configured for analog mode.
Note: P2.2-P2.7 are available on 40-pin and 32-pin packages.		

SFR Definition 19.22. P2MDOUT: Port 2 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P2MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xA6; SFR Page = 0x0F

Bit	Name	Function
7:0	P2MDOUT[7:0]	**Output Configuration Bits for P2.7–P2.0 (respectively).** These bits are ignored if the corresponding bit in register P2MDIN is logic 0. 0: Corresponding P2.n Output is open-drain. 1: Corresponding P2.n Output is push-pull.
Note: P2.2-P2.7 are available on 40-pin and 32-pin packages.		

SFR Definition 19.25. P3MDIN: Port 3 Input Mode

Bit	7	6	5	4	3	2	1	0
Name	P3MDIN[7:0]							
Type	R/W							
Reset	1	1	1	1	1	1	1	1

SFR Address = 0xF4; SFR Page = 0x0F

Bit	Name	Function
7:0	P3MDIN[7:0]	**Analog Configuration Bits for P3.7–P3.0 (respectively).** Port pins configured for analog mode have their weak pull-up and digital receiver disabled. For analog mode, the pin also needs to be configured for open-drain mode in the P3MDOUT register. 0: Corresponding P3.n pin is configured for analog mode. 1: Corresponding P3.n pin is not configured for analog mode.
Note: P3.0 is available on 40-pin and 32-pin packages. P3.1-P3.7 are available on 40-pin packages		

SFR Definition 19.26. P3MDOUT: Port 3 Output Mode

Bit	7	6	5	4	3	2	1	0
Name	P3MDOUT[7:0]							
Type	R/W							
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xAE; SFR Page = 0x0F

Bit	Name	Function
7:0	P3MDOUT[7:0]	**Output Configuration Bits for P3.7–P3.0 (respectively).** These bits are ignored if the corresponding bit in register P3MDIN is logic 0. 0: Corresponding P3.n Output is open-drain. 1: Corresponding P3.n Output is push-pull.
Note: P3.0 is available on 40-pin and 32-pin packages. P3.1-P3.7 are available on 40-pin packages		

3. The LIN controller does not directly support LIN Version 1.3 Extended Frames. If the application detects an unknown identifier (e.g. extended identifier), it has to write a 1 to the STOP bit (LIN0CTRL.7) instead of setting the DTACK (LIN0CTRL.4) bit. At that time, steps 2 through 5 can then be skipped. In this situation, the LIN controller stops the processing of LIN communication until the next SYNC BREAK is received.
4. Changing the configuration of the checksum during a transaction will cause the interface to reset and the transaction to be lost. To prevent this, the checksum should not be configured while a transaction is in progress. The same applies to changes in the LIN interface mode from slave mode to master mode and from master mode to slave mode.

20.5. Sleep Mode and Wake-Up

To reduce the system's power consumption, the LIN Protocol Specification defines a Sleep Mode. The message used to broadcast a Sleep Mode request must be transmitted by the LIN master application in the same way as a normal transmit message. The LIN slave application must decode the Sleep Mode Frame from the Identifier and data bytes. After that, it has to put the LIN slave node into the Sleep Mode by setting the SLEEP bit (LIN0CTRL.6).

If the SLEEP bit (LIN0CTRL.6) of the LIN slave application is not set and there is no bus activity for four seconds (specified bus idle timeout), the IDLTOUT bit (LIN0ST.6) is set and an interrupt request is generated. After that the application may assume that the LIN bus is in Sleep Mode and set the SLEEP bit (LIN0CTRL.6).

Sending a wake-up signal from the master or any slave node terminates the Sleep Mode of the LIN bus. To send a wake-up signal, the application has to set the WUPREQ bit (LIN0CTRL.1). After successful transmission of the wake-up signal, the DONE bit (LIN0ST.0) of the master node is set and an interrupt request is generated. The LIN slave does not generate an interrupt request after successful transmission of the wake-up signal but it generates an interrupt request if the master does not respond to the wake-up signal within 150 milliseconds. In that case, the ERROR bit (LIN0ST.2) and TOUT bit (LIN0ERR.2) are set. The application then has to decide whether or not to transmit another wake-up signal.

All LIN nodes that detect a wake-up signal will set the WAKEUP (LIN0ST.1) and DONE bits (LIN0ST.0) and generate an interrupt request. After that, the application has to clear the SLEEP bit (LIN0CTRL.6) in the LIN slave.

20.6. Error Detection and Handling

The LIN controller generates an interrupt request and stops the processing of the current frame if it detects an error. The application has to check the type of error by processing LIN0ERR. After that, it has to reset the error register and the ERROR bit (LIN0ST.2) by writing a 1 to the RSTERR bit (LIN0CTRL.2). Starting a new message with the LIN controller selected as master or sending a Wakeup signal with the LIN controller selected as a master or slave is possible only if the ERROR bit (LIN0ST.2) is set to 0.

Table 22.1. SMBus Clock Source Selection

SMBCS1	SMBCS0	SMBus Clock Source
0	0	Timer 0 Overflow
0	1	Timer 1 Overflow
1	0	Timer 2 High Byte Overflow
1	1	Timer 2 Low Byte Overflow

The SMBCS1–0 bits select the SMBus clock source, which is used only when operating as a master or when the Free Timeout detection is enabled. When operating as a master, overflows from the selected source determine the absolute minimum SCL low and high times as defined in Equation 22.1. Note that the selected clock source may be shared by other peripherals so long as the timer is left running at all times. For example, Timer 1 overflows may generate the SMBus and UART baud rates simultaneously. Timer configuration is covered in Section “25. Timers” on page 259.

$$T_{HighMin} = T_{LowMin} = \frac{1}{f_{ClockSourceOverflow}}$$

Equation 22.1. Minimum SCL High and Low Times

The selected clock source should be configured to establish the minimum SCL High and Low times as per Equation 22.1. When the interface is operating as a master (and SCL is not driven or extended by any other devices on the bus), the typical SMBus bit rate is approximated by Equation 22.2.

$$BitRate = \frac{f_{ClockSourceOverflow}}{3}$$

Equation 22.2. Typical SMBus Bit Rate

Figure 22.4 shows the typical SCL generation described by Equation 22.2. Notice that T_{HIGH} is typically twice as large as T_{LOW}. The actual SCL output may vary due to other devices on the bus (SCL may be extended low by slower slave devices, or driven low by contending master devices). The bit rate when operating as a master will never exceed the limits defined by equation Equation 22.1.

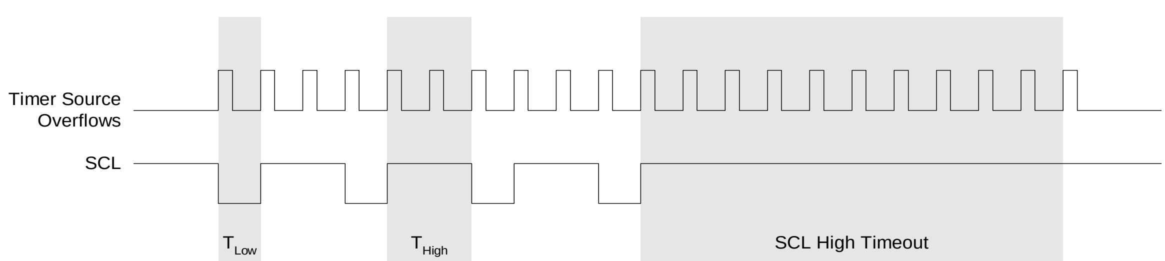

Figure 22.4. Typical SMBus SCL Generation

SFR Definition 22.1. SMB0CF: SMBus Clock/Configuration

Bit	7	6	5	4	3	2	1	0
Name	ENSMB	INH	BUSY	EXTHOLD	SMBTOE	SMBFTE	SMBCS[1:0]	
Type	R/W	R/W	R	R/W	R/W	R/W	R/W	
Reset	0	0	0	0	0	0	0	0

SFR Address = 0xC1; SFR Page = 0x00

Bit	Name	Function
7	ENSMB	**SMBus Enable.** This bit enables the SMBus interface when set to 1. When enabled, the interface constantly monitors the SDA and SCL pins.
6	INH	**SMBus Slave Inhibit.** When this bit is set to logic 1, the SMBus does not generate an interrupt when slave events occur. This effectively removes the SMBus slave from the bus. Master Mode interrupts are not affected.
5	BUSY	**SMBus Busy Indicator.** This bit is set to logic 1 by hardware when a transfer is in progress. It is cleared to logic 0 when a STOP or free-timeout is sensed.
4	EXTHOLD	**SMBus Setup and Hold Time Extension Enable.** This bit controls the SDA setup and hold times according to Table 22.2. 0: SDA Extended Setup and Hold Times disabled. 1: SDA Extended Setup and Hold Times enabled.
3	SMBTOE	**SMBus SCL Timeout Detection Enable.** This bit enables SCL low timeout detection. If set to logic 1, the SMBus forces Timer 3 to reload while SCL is high and allows Timer 3 to count when SCL goes low. If Timer 3 is configured to Split Mode, only the High Byte of the timer is held in reload while SCL is high. Timer 3 should be programmed to generate interrupts at 25 ms, and the Timer 3 interrupt service routine should reset SMBus communication.
2	SMBFTE	**SMBus Free Timeout Detection Enable.** When this bit is set to logic 1, the bus will be considered free if SCL and SDA remain high for more than 10 SMBus clock source periods.
1:0	SMBCS[1:0]	**SMBus Clock Source Selection.** These two bits select the SMBus clock source, which is used to generate the SMBus bit rate. The selected device should be configured according to Equation 22.1. 00: Timer 0 Overflow 01: Timer 1 Overflow 10:Timer 2 High Byte Overflow 11: Timer 2 Low Byte Overflow

22.5.2. Read Sequence (Master)

During a read sequence, an SMBus master reads data from a slave device. The master in this transfer will be a transmitter during the address byte, and a receiver during all data bytes. The SMBus interface generates the START condition and transmits the first byte containing the address of the target slave and the data direction bit. In this case the data direction bit (R/W) will be logic 1 (READ). Serial data is then received from the slave on SDA while the SMBus outputs the serial clock. The slave transmits one or more bytes of serial data. An interrupt is generated after each received byte.

Software must write the ACK bit at that time to ACK or NACK the received byte. Writing a 1 to the ACK bit generates an ACK; writing a 0 generates a NACK. Software should write a 0 to the ACK bit for the last data transfer, to transmit a NACK. The interface exits Master Receiver Mode after the STO bit is set and a STOP is generated. The interface will switch to Master Transmitter Mode if SMB0DAT is written while an active Master Receiver. Figure 22.6 shows a typical master read sequence. Two received data bytes are shown, though any number of bytes may be received. Notice that the ‘data byte transferred’ interrupts occur **before** the ACK cycle in this mode.

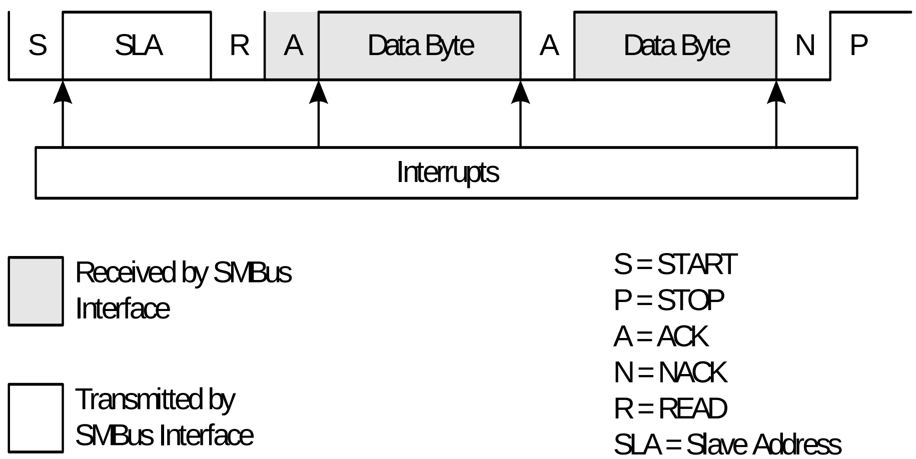

Figure 22.6. Typical Master Read Sequence

24. Enhanced Serial Peripheral Interface (SPI0)

The Enhanced Serial Peripheral Interface (SPI0) provides access to a flexible, full-duplex synchronous serial bus. SPI0 can operate as a master or slave device in both 3-wire or 4-wire modes, and supports multiple masters and slaves on a single SPI bus. The slave-select (NSS) signal can be configured as an input to select SPI0 in slave mode, or to disable Master Mode operation in a multi-master environment, avoiding contention on the SPI bus when more than one master attempts simultaneous data transfers. NSS can also be configured as a chip-select output in master mode, or disabled for 3-wire operation. Additional general purpose port I/O pins can be used to select multiple slave devices in master mode.

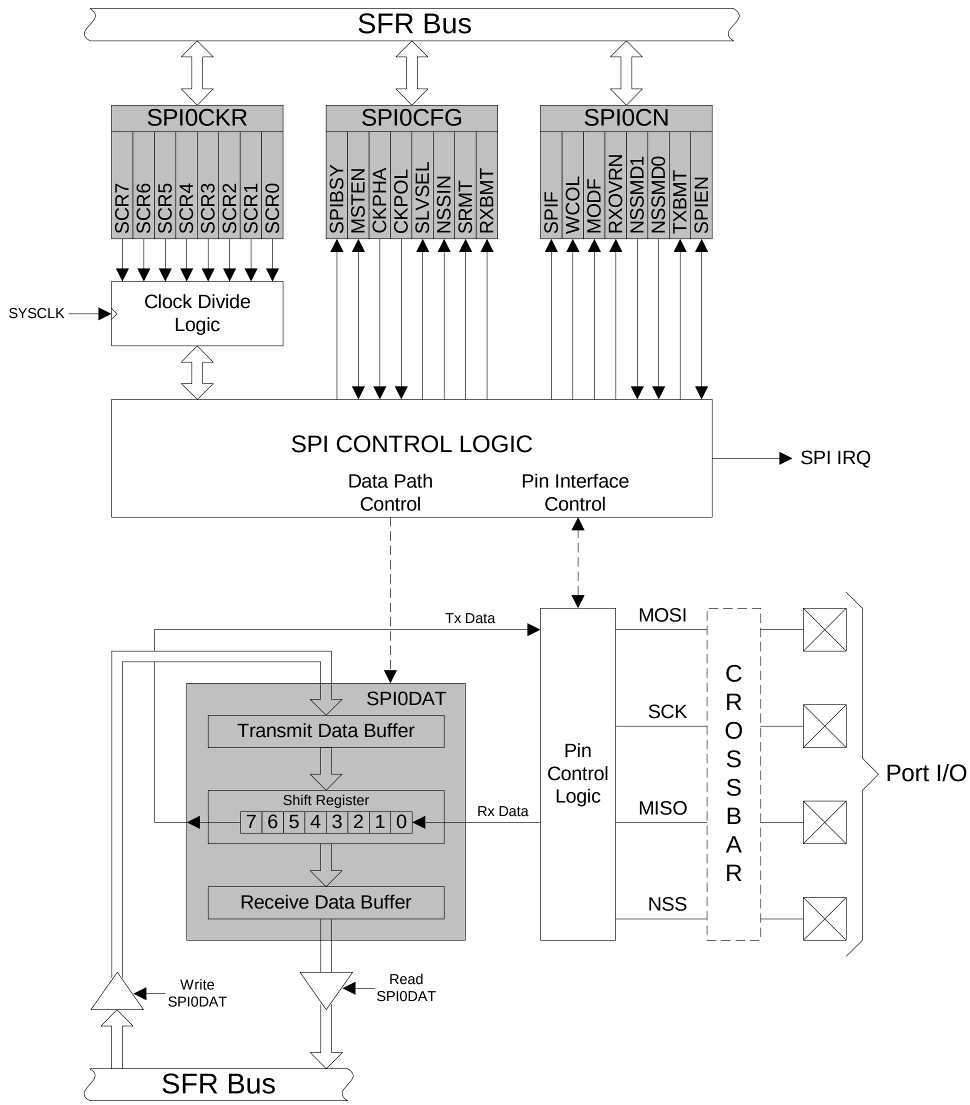

Figure 24.1. SPI Block Diagram

Table 26.2. PCA0CPM and PCA0PWM Bit Settings for PCA Capture/Compare Modules

Operational Mode	PCA0CPMn								PCA0PWM				
Bit Number	7	6	5	4	3	2	1	0	7	6	5	4–2	1–0
Capture triggered by positive edge on CEXn	X	X	1	0	0	0	0	A	0	X	B	XXX	XX
Capture triggered by negative edge on CEXn	X	X	0	1	0	0	0	A	0	X	B	XXX	XX
Capture triggered by any transition on CEXn	X	X	1	1	0	0	0	A	0	X	B	XXX	XX
Software Timer	X	C	0	0	1	0	0	A	0	X	B	XXX	XX
High Speed Output	X	C	0	0	1	1	0	A	0	X	B	XXX	XX
Frequency Output	X	C	0	0	0	1	1	A	0	X	B	XXX	XX
8-Bit Pulse Width Modulator (7)	0	C	0	0	E	0	1	A	0	X	B	XXX	00
9-Bit Pulse Width Modulator (7)	0	C	0	0	E	0	1	A	D	X	B	XXX	01
10-Bit Pulse Width Modulator (7)	0	C	0	0	E	0	1	A	D	X	B	XXX	10
11-Bit Pulse Width Modulator (7)	0	C	0	0	E	0	1	A	D	X	B	XXX	11
16-Bit Pulse Width Modulator	1	C	0	0	E	0	1	A	0	X	B	XXX	XX

Notes:
1. X = Don’t Care (no functional difference for individual module if 1 or 0).
2. A = Enable interrupts for this module (PCA interrupt triggered on CCFn set to 1).
3. B = Enable 8th, 9th, 10th or 11th bit overflow interrupt (Depends on setting of CLSEL[1:0]).
4. C = When set to 0, the digital comparator is off. For high speed and frequency output modes, the associated pin will not toggle. In any of the PWM modes, this generates a 0% duty cycle (output = 0).
5. D = Selects whether the Capture/Compare register (0) or the Auto-Reload register (1) for the associated channel is accessed via addresses PCA0CPHn and PCA0CPLn.
6. E = When set, a match event will cause the CCFn flag for the associated channel to be set.
7. All modules set to 8, 9, 10 or 11-bit PWM mode use the same cycle length setting.

26.3.1. Edge-triggered Capture Mode

In this mode, a valid transition on the CEXn pin causes the PCA to capture the value of the PCA counter/timer and load it into the corresponding module's 16-bit capture/compare register (PCA0CPLn and PCA0CPHn). The CAPPn and CAPNn bits in the PCA0CPMn register are used to select the type of transition that triggers the capture: low-to-high transition (positive edge), high-to-low transition (negative edge), or either transition (positive or negative edge). When a capture occurs, the Capture/Compare Flag (CCFn) in PCA0CN is set to logic 1. An interrupt request is generated if the CCFn interrupt for that module is enabled. The CCFn bit is not automatically cleared by hardware when the CPU vectors to the interrupt service routine, and must be cleared by software. If both CAPPn and CAPNn bits are set to logic 1, then the state of the Port pin associated with CEXn can be read directly to determine whether a rising-edge or falling-edge caused the capture.

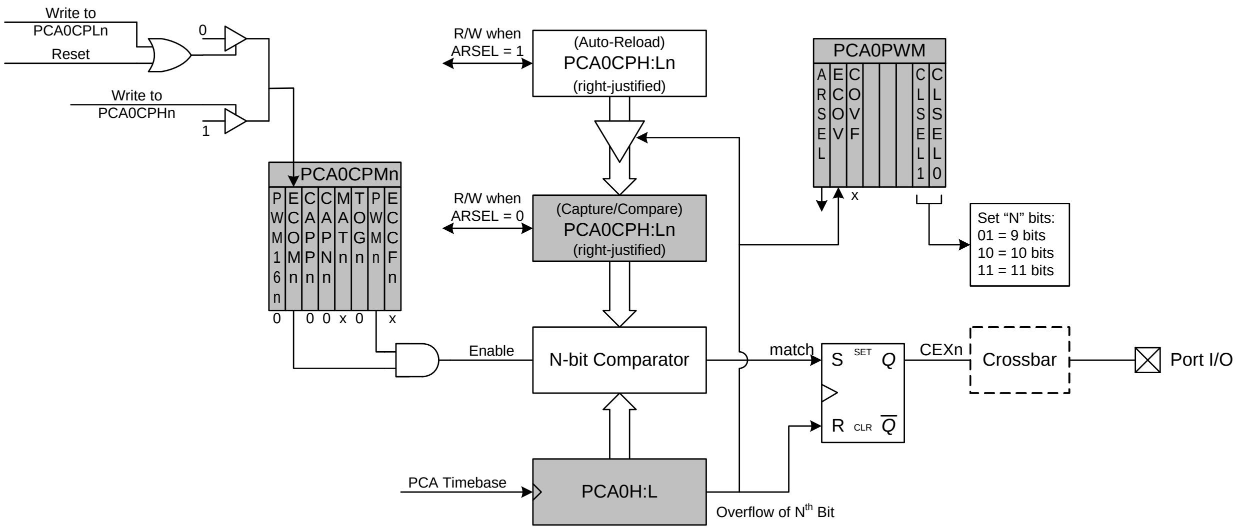

Figure 26.9. PCA 9, 10 and 11-Bit PWM Mode Diagram

26.3.6. 16-Bit Pulse Width Modulator Mode

A PCA module may also be operated in 16-Bit PWM mode. 16-bit PWM mode is independent of the other (8/9/10/11-bit) PWM modes. In this mode, the 16-bit capture/compare module defines the number of PCA clocks for the low time of the PWM signal. When the PCA counter matches the module contents, the output on CEXn is asserted high; when the 16-bit counter overflows, CEXn is asserted low. To output a varying duty cycle, new value writes should be synchronized with PCA CCFn match interrupts. 16-Bit PWM Mode is enabled by setting the ECOMn, PWMn, and PWM16n bits in the PCA0CPMn register. For a varying duty cycle, match interrupts should be enabled (ECCFn = 1 AND MATn = 1) to help synchronize the capture/compare register writes. If the MATn bit is set to 1, the CCFn flag for the module will be set each time a 16-bit comparator match (rising edge) occurs. The CF flag in PCA0CN can be used to detect the overflow (falling edge). The duty cycle for 16-Bit PWM Mode is given by Equation 26.4.

Important Note About Capture/Compare Registers: When writing a 16-bit value to the PCA0 Capture/Compare registers, the low byte should always be written first. Writing to PCA0CPLn clears the ECOMn bit to 0; writing to PCA0CPHn sets ECOMn to 1.

$$\text{Duty Cycle} = \frac{(65536 - \text{PCA0CPn})}{65536}$$

Equation 26.4. 16-Bit PWM Duty Cycle

Using Equation 26.4, the largest duty cycle is 100% (PCA0CPn = 0), and the smallest duty cycle is 0.0015% (PCA0CPn = 0xFFFF). A 0% duty cycle may be generated by clearing the ECOMn bit to 0.